Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DSA in 75 Days Coding Challenge

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 1 : Longest Palindromic Substring

To find the longest palindromic substring within a given string, you need to identify the longest substring that reads the same forwards and backwards.

Example:

Given the string "babad", the longest palindromic substring is "bab" or "aba".

DAY 2 : Roman To Integer

The Roman to Integer problem involves converting a Roman numeral string into an integer.

Example:

Given the Roman numeral "IV", the corresponding integer is 4.

DAY 3 : Power of Two

The Power of Two problem involves determining whether a given integer is a power of two.

Example:

For integer 8, the answer is true because 8 is equal to 2 raised to the power of 3 (2^3).

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 4 : Reverse Words in a String

To reverse the words in a string, you would typically split the string into words, reverse the order of the words, and then join them back together.

Example:

If the input string is "Hello world", the output would be "world Hello".

Here's how it works:

1. Split the string into words: ["Hello", "world"]
2. Reverse the order of the words: ["world", "Hello"]
3. Join the words back together with spaces: "world Hello"

DAY 5 : Longest Common Prefix

The longest common prefix of a set of strings is the longest string that is a prefix of all strings in the set.

Example:

For the set of strings: ["flower", "flow", "flight"], the longest common prefix is "fl".

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 6 : Valid Anagram

A valid anagram is a word or phrase formed by rearranging the letters of another word or phrase.

Example:

"listen" and "silent" are valid anagrams because they have the same letters, just rearranged.

DAY 7 : Best Time to buy and sell stock

The best time to buy and sell stocks refers to finding the optimal time to buy low and sell high to maximize profit within a given period.

Example:

Suppose the stock prices over a week are: [7, 1, 5, 3, 6, 4]

The best time to buy would be when the price is lowest, at $1 on the second day, and the best time to sell would be when the price is highest, at $6 on the fifth day. So, buying on the second day and selling on the fifth day would yield the maximum profit, which is $5.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 8 : Maximum Subarray

The maximum subarray refers to the contiguous subarray within an array that has the largest sum of its elements.

Example:

Consider an array: [-2, 1, -3, 4, -1, 2, 1, -5, 4]

The maximum subarray in this array is [4, -1, 2, 1], with a sum of 6. This subarray has the largest sum among all possible contiguous subarrays within the given array.

DAY 9 : Middle of the Linked List

The "Middle of the Linked List" problem involves finding the middle node of a singly linked list.

Example:

Given a linked list: 1 -> 2 -> 3 -> 4 -> 5

The middle node is node 3 (the third node) in this case. If the linked list has an even number of nodes, there are two middle nodes, and you need to return the second middle node.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 10 : Palindrome Linked List

The "Palindrome Linked List" problem involves determining whether a singly linked list is a palindrome.

Example:

Given a linked list: 1 -> 2 -> 2 -> 1

The linked list is a palindrome because it reads the same forwards and backwards.

DAY 11 : Median of two sorted Arrays

The "Median of Two Sorted Arrays" problem involves finding the median of the two sorted arrays.

Example:

Given two sorted arrays [1, 3] and [2], the median is 2.

DAY 12 : LRU Cache

The "LRU Cache" problem involves designing a data structure that supports the Least Recently Used (LRU) cache eviction policy.


Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 13 : LFU Cache

The "LFU Cache" problem involves designing a data structure with a fixed capacity that supports the Least Frequently Used (LFU) cache eviction policy.

DAY 14 : Different ways to add Parentheses

The "Different Ways to Add Parentheses" problem involves evaluating different ways of adding parentheses to an arithmetic expression and computing all possible values.

Example:

Given the expression "2-1-1", the possible ways to add parentheses and their resulting values are:

- (2-(1-1)) = 2
- ((2-1)-1) = 0

DAY 15 : Sudoku Solver

The Sudoku Solver problem involves completing a partially filled 9x9 Sudoku grid such that every row, column, and 3x3 subgrid contains all numbers from 1 to 9.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 16 : Combination Sum II

The Combination Sum II problem involves finding all unique combinations of a set of candidate numbers that sum up to a target value, where each number in the combination may only be used once.

Example:

Given candidate set [10, 1, 2, 7, 6, 1, 5] and target 8, a solution might be [[1, 1, 6], [1, 2, 5], [1, 7], [2, 6]].

DAY 17 : Add Two Numbers

The "Add Two Numbers" problem involves adding two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return it as a linked list.

Example:

Given two linked lists representing the numbers 243 and 564 (stored in reverse order), the sum would be 807. So, the resulting linked list would be: 7 -> 0 -> 8

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 18 : Implement Stack using Queues

The "Implement Stack using Queues" problem involves designing a stack data structure using only queues.

Example:

Implementing a stack using queues means designing a data structure where the last element added to the stack is the first to be removed, similar to how a stack behaves.

DAY 19 : Top K Frequent Elements

The "Top K Frequent Elements" problem involves finding the k most frequent elements in an integer array.

Example:

Given an array [1, 1, 1, 2, 2, 3] and k = 2, the top 2 frequent elements are 1 and 2.

DAY 20 : Min Stack

The "Min Stack" problem involves designing a stack data structure that supports push, pop, top, and retrieving the minimum element in constant time.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 21 : Kosaraju algorithm O(N)

The "Kosaraju's Algorithm" problem involves finding strongly connected components (SCCs) in a directed graph efficiently, with a time complexity of O(N), where N is the number of vertices in the graph.

Example:

Given a directed graph, Kosaraju's algorithm can efficiently find all strongly connected components in linear time.

DAY 22 : Implementing Dijkstra Algorithm

The "Dijkstra's Algorithm" problem involves finding the shortest path from a single source vertex to all other vertices in a weighted graph with non-negative weights.

Example:

Given a weighted graph and a source vertex, Dijkstra's algorithm efficiently finds the shortest paths from the source vertex to all other vertices.

DAY 23 : Same Tree

The "Same Tree" problem involves determining whether two binary trees are identical in structure and node values.


Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 24 : Binary Tree ZigZag Order Traversal

The "Binary Tree Zigzag Level Order Traversal" problem involves traversing a binary tree level by level in a zigzag pattern, switching between left-to-right and right-to-left traversal at each level.

DAY 25 : Lowest Common Ancestor Of a Binary Tree

The "Lowest Common Ancestor of a Binary Tree" problem involves finding the lowest common ancestor (LCA) of two given nodes in a binary tree.

DAY 26 : Maximum Width Of a Binary Tree

The "Maximum Width of a Binary Tree" problem involves finding the maximum width of a binary tree, which is the maximum number of nodes in any level of the tree.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 27 : Vertical Order Traversal Of a Binary Tree

The "Vertical Order Traversal of a Binary Tree" problem involves traversing a binary tree vertically, from top to bottom and from left to right within each vertical level.

DAY 28 : Maximum Depth of Binary Tree

The "Maximum Depth of Binary Tree" problem involves finding the maximum depth of a binary tree, which is the number of nodes along the longest path from the root node down to the farthest leaf node.

DAY 29 : Balanced Binary Tree

The "Balanced Binary Tree" problem involves determining whether a binary tree is balanced, meaning the heights of the two subtrees of any node never differ by more than one.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 30 : Binary Tree Level Order Traversal

The "Binary Tree Level Order Traversal" problem involves traversing a binary tree level by level, from left to right.

DAY 31 : Binary Tree Level Order Traversal

The "Binary Tree Level Order Traversal" problem involves traversing a binary tree level by level, from left to right.

DAY 32 : Maximum Width Of Binary Tree

The "Maximum Width of Binary Tree" problem involves finding the maximum width of a binary tree, which is the maximum number of nodes in any level of the tree.

DAY 33 : Binary Tree Postorder Traversal

The "Binary Tree Postorder Traversal" problem involves traversing a binary tree in a postorder manner, meaning you visit the left subtree, then the right subtree, and finally the current node.


Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 34 : Binary Tree Preorder Traversal

The "Binary Tree Preorder Traversal" problem involves traversing a binary tree in a preorder manner, meaning you visit the current node, then the left subtree, and finally the right subtree.

DAY 35 : Binary Tree Inorder Traversal

The "Binary Tree Inorder Traversal" problem involves traversing a binary tree in an inorder manner, meaning you visit the left subtree, then the current node, and finally the right subtree.

DAY 36 : Rotting Oranges

The "Rotting Oranges" problem involves determining the minimum time required to rot all oranges in a grid, given that rotten oranges can rot their adjacent fresh oranges.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 37 : Sliding Window Maximum

The "Sliding Window Maximum" problem involves finding the maximum value in each sliding window of a given array.

Example:

Given an array [1, 3, -1, -3, 5, 3, 6, 7] and a window size of 3, the maximum values in each window would be [3, 3, 5, 5, 6, 7].

DAY 38 : Course Schedule

The "Course Schedule" problem involves determining whether it is possible to complete all courses in a given list, where each course has prerequisites.

Example:

Given the total number of courses and a list of prerequisite pairs:

2, [[1,0]]

It is possible to complete both courses, where course 1 is a prerequisite of course 0.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 39 : Largest Rectangle in Histogram

The "Largest Rectangle in Histogram" problem involves finding the largest rectangular area that can be formed by a histogram.

Example:

Given a histogram [2, 1, 5, 6, 2, 3], the largest rectangle that can be formed has an area of 10, which corresponds to the bars with heights 5 and 6.

DAY 40 : Topological Sort DFS

The "Topological Sort using Depth-First Search (DFS)" problem involves ordering a directed graph's vertices in such a way that for every directed edge $u \rightarrow v$, vertex u comes before vertex v in the ordering.

DAY 41 : Topological Sort BFS

The "Topological Sort using Breadth-First Search (BFS)" problem involves ordering a directed graph's vertices in such a way that for every directed edge $u \rightarrow v$, vertex u comes before vertex v in the ordering.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 42 : Valid Parentheses

The "Valid Parentheses" problem involves determining whether a given string of parentheses is valid.

Example:

For the string "([{}])", the parentheses are valid because each opening parenthesis has a corresponding closing parenthesis in the correct order.

DAY 43 : Implement Queue using stacks

The "Implement Queue using Stacks" problem involves designing a queue data structure using only stacks.

Example:

Implementing a queue using stacks means designing a data structure where the first element added to the queue is the first to be removed, similar to how a queue behaves.

DAY 44 : Minimum Path Sum

The "Minimum Path Sum" problem involves finding the minimum sum of a path from the top-left corner to the bottom-right corner of a grid, moving only right or down.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 45 : Coin Change

The "Coin Change" problem involves finding the minimum number of coins required to make a certain amount of money, given a set of coin denominations.

Example:

Given coins of denominations [1, 2, 5] and a target amount of 11, the minimum number of coins required is 3 (one 5-cent coin and two 3-cent coins).

DAY 46 : Matrix Chain Multiplication

The "Matrix Chain Multiplication" problem involves finding the most efficient way to multiply a given sequence of matrices.

Example:

Given a sequence of matrices A, B, C, and D with dimensions:

- A: 10x30
- B: 30x5
- C: 5x60
- D: 60x10

The most efficient way to multiply these matrices would result in the minimum number of scalar multiplications.


Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 47 : Edit Distance

The "Edit Distance" problem involves determining the minimum number of operations required to convert one string into another, where the allowed operations are insertion, deletion, or substitution of a character.

Example:

Given two strings "horse" and "ros", the edit distance is 3 because the minimum operations required are:

1. Delete 'h' from "horse" to get "orse".
2. Replace 'r' with 'o' in "orse" to get "ose".
3. Add 's' to the end of "ose" to get "ros".

DAY 48 : Egg Dropping Puzzle

The "Egg Dropping Puzzle" problem involves finding the minimum number of trials needed to find the critical floor, where the critical floor is the highest floor from which an egg dropped will break.

Example:

Given a building with 100 floors and two eggs, the minimum number of trials needed to find the critical floor using the optimal strategy is 14.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 49 : Concatenation of Array

Concatenation of arrays refers to the process of combining two arrays into a single array by appending all elements of one array to the end of another array.

Example:

Given an array of numbers:
Array 1: [1, 2, 3]
Array 2: [4, 5, 6]
Concatenating these arrays results in a new array:
Concatenated Array: [1, 2, 3, 4, 5, 6]

Here, all elements from Array 2 are appended to the end of Array 1, creating a new array with combined elements.

DAY 50 : Number Of Good Pairs

The number of good pairs in an array counts pairs of indices where the elements are equal and the first index is less than the second.

Example:

Array: [1, 2, 3, 2, 2, 1, 3]

Number of good pairs: 4


Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 51 : Pascal’s Triangle

Pascal's triangle is a triangular array of binomial coefficients. Each number in the triangle is the sum of the two directly above it.

Example:

Consider the first 5 rows of Pascal's triangle:

```
1
  1 1
 1 2 1
1 3 3 1
1 4 6 4 1
```

In Pascal's triangle, each number is the sum of the two numbers directly above it. For instance, the number 6 in the fourth row is the sum of 3 and 3 in the row above it.

DAY 52 : Subsets II

The Subsets II problem involves generating all possible unique subsets from a collection of integers, while considering duplicate elements.

Example:

Given a set of integers with duplicates like [1, 2, 2], the unique subsets would be [[], [1], [2], [1, 2], [2, 2], [1, 2, 2]].

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 53 : Palindrome Partitioning

The Palindrome Partitioning problem involves partitioning a string into substrings such that each substring is a palindrome.

Example:

Given the string "aab", possible palindrome partitionings could be ["a", "a", "b"] and ["aa", "b"].

DAY 54 : Shuffle the Array

Shuffling an array refers to the process of randomly rearranging the elements within the array.

Example:

Given an array of numbers:
Array: [1, 2, 3, 4, 5]
After shuffling the array, it might become:
Shuffled Array: [3, 5, 1, 4, 2]

Here, the elements within the array are randomly rearranged to create a new arrangement, resulting in a shuffled array.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 55 : Repeated String Match

Repeated String Match is a problem in which you're given two strings, A and B, and you need to determine the minimum number of times you should repeat string A so that string B becomes a substring of the repeated A.

Example:

Given strings A = "abcd" and B = "cdabcdab", you might need to repeat A twice to get "abcdabcd", and then B becomes a substring of it. So, the minimum number of repetitions of A is 2 in this case.

DAY 56 : Merge two sorted Lists

The "Merge Two Sorted Lists" problem involves merging two sorted linked lists into a single sorted linked list.

Example:

Given two sorted linked lists:

List 1: 1 -> 2 -> 4

List 2: 1 -> 3 -> 4

The merged sorted linked list would be: 1 -> 1 -> 2 -> 3 -> 4 -> 4

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 57 : Linked List Cycle

The "Linked List Cycle" problem involves determining whether a linked list has a cycle (i.e., whether a node in the list points to a previous node, causing a loop).

Example:

Given a linked list: 1 -> 2 -> 3 -> 4 -> 5 -> 2 (points back to node 2)

The linked list has a cycle.

DAY 58 : 0-1 Knapsack Problem

The "0-1 Knapsack Problem" involves finding the maximum value that can be obtained by selecting a subset of items with given weights and values, such that the sum of the weights of the selected items is less than or equal to a given weight capacity.

Example:

Given a knapsack with a weight capacity of 10 and the following items:

- Item 1: Weight 5, Value 10
- Item 2: Weight 4, Value 40
- Item 3: Weight 6, Value 30

The maximum value that can be obtained is 70 by selecting items 1 and 2.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 59 : Minimum Spanning Tree

The "Minimum Spanning Tree (MST)" problem involves finding a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together with the minimum possible total edge weight.

Example:

Given an edge-weighted graph, the Minimum Spanning Tree algorithm efficiently finds a spanning tree with the minimum total edge weight.

DAY 60 : Maximum Product Subarray

The "Maximum Product Subarray" problem involves finding the contiguous subarray within an array containing at least one number which has the largest product.

Example:

Given an array [2, 3, -2, 4], the contiguous subarray [2, 3] has the largest product of 6.


Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 61 : Bellman-Ford Algorithm

The "Bellman-Ford Algorithm" problem involves finding the shortest path from a single source vertex to all other vertices in a weighted graph, even when the graph contains negative weight edges.

Example:

Given a weighted graph with possibly negative weight edges and a source vertex, the Bellman-Ford algorithm can efficiently find the shortest paths from the source vertex to all other vertices.

DAY 62 : Longest Common Subsequence

The "Longest Common Subsequence" problem involves finding the longest subsequence that is common to two given sequences.

Example:

Given two sequences "abcde" and "ace", the longest common subsequence is "ace" with a length of 3.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 63 : Partition Equal Subset Sum

The "Partition Equal Subset Sum" problem involves determining whether a given set can be partitioned into two subsets with equal sum.

Example:

Given a set [1, 5, 11, 5], it can be partitioned into two subsets [1, 5, 5] and [11] with equal sum, making the answer true.

DAY 64 : Job Sequencing Problem

The "Job Sequencing Problem" involves finding the maximum profit sequence of jobs with given deadlines and profits.

Example:

Given jobs with deadlines and profits:
Job 1: Deadline 2, Profit 100
Job 2: Deadline 1, Profit 50
Job 3: Deadline 2, Profit 10
The maximum profit sequence would be Job 1 followed by Job 3.


Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 65 : Maximum XOR of Two Numbers in an Array

The "Maximum XOR of Two Numbers in an Array" problem involves finding the maximum XOR value of any two integers in a given array.

Example:

Given an array [3, 10, 5, 25, 2, 8], the maximum XOR value of any two integers is 28 (resulting from XOR of 5 and 25).

DAY 66 : Reverse Linked List

The "Reverse Linked List" problem involves reversing a singly linked list.

Example:

Given a linked list: 1 -> 2 -> 3 -> 4 -> 5

The reversed linked list would be: 5 -> 4 -> 3 -> 2 -> 1

DAY 67 : Power Set

The "Power Set" problem involves finding all possible subsets of a given set, including the empty set and the set itself.

Example:

Given a set {1, 2, 3}, the power set would be {{}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}}.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 68 : Implement Trie (Prefix Tree)

The "Implement Trie (Prefix Tree)" problem involves designing a trie data structure to store a set of words and efficiently support operations like search, insert, and startsWith.

Example:

Given a set of words {"apple", "banana", "orange"}, a trie can efficiently store these words and support operations like searching for a word or finding words with a specific prefix.

DAY 69 : Word Break

The "Word Break" problem involves determining whether a given string can be segmented into a space-separated sequence of one or more dictionary words.

Example:

Given a string "leetcode" and a dictionary ["leet", "code"], the string can be segmented into "leet" and "code", making the answer true.

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 70 : Search in Rotated Sorted Array

The "Search in Rotated Sorted Array" problem involves finding the target value in a rotated sorted array.

Example:

Given the rotated sorted array [4, 5, 6, 7, 0, 1, 2] and target value 0, the index of the target value is 4.

DAY 71 : Kth Largest Element in an Array

The "Kth Largest Element in an Array" problem involves finding the kth largest element in an unsorted array.

Example:

Given the array [3, 2, 1, 5, 6, 4] and k = 2, the 2nd largest element is 5.

DAY 72 : Palindromic Partitioning

The "Palindromic Partitioning" problem involves partitioning a string into substrings such that each substring is a palindrome.

Example:

Given a string "aab", the possible palindromic partitionings are ["a", "a", "b"] and ["aa", "b"].

Omkar
@omsrivastava

Helps job aspirants to crack FAANG product companies

DAY 73 : Minimum Cost to cut a Stick

The "Minimum Cost to Cut a Stick" problem involves finding the minimum cost required to cut a stick into pieces of given lengths.

Example:

Given a stick of length 8 and the cost of cutting at positions [1, 2, 3, 4, 5, 6, 7], the minimum cost to cut the stick into pieces of lengths [1, 2, 3, 4, 5, 6, 7, 8] is 31.

DAY 74 : Single Element in a Sorted Array

The "Single Element in a Sorted Array" problem involves finding the single element in a sorted array where every element appears twice, except for one element which appears only once.

Example:

Given the sorted array [1, 1, 2, 3, 3, 4, 4, 5, 5], the single element is 2.

Omkar
@omsrivastava

Helps job aspirants to crack
FAANG product companies

DAY 75 : Find Median from Data Stream

The "Find Median from Data Stream" problem involves designing a data structure to support finding the median of elements in a data stream.

Example:

Given a stream of integers: [1, 2, 3, 4, 5]
The median after each insertion would be: [1, 1.5, 2, 2.5, 3]

DAY 75 : Floyd Warshall

The "Floyd-Warshall Algorithm" problem involves finding the shortest paths between all pairs of vertices in a weighted graph, including negative weight edges.

Example:

Given a weighted graph, the Floyd-Warshall algorithm efficiently finds the shortest paths between all pairs of vertices.